

УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ И ЭКОНОМИЧЕСКИХ СИСТЕМАХ

УДК 004.4'233

СОВЕРШЕНСТВОВАНИЕ МЕТОДИКИ УПРАВЛЕНИЯ ПРОЕКТАМИ ИТ-КОМПАНИИ НА ОСНОВЕ МОДЕЛИ СММ

Статья поступила в редакцию 06.12.2016, в окончательном варианте – 26.01.2017.

Мунтянова Татьяна Петровна, аспирант, Астраханский государственный технический университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 16, e-mail: orekhova.t.p@gmail.com

Разработана теоретико-множественная модель, отражающая деятельность ИТ-компании на основе учета системного жизненного цикла проекта, связанного с разработкой программного обеспечения. Описана структурно-функциональная модель управления ИТ-проектами. Проанализированы различные унифицированные подходы к управлению ИТ-проектами, выбрана наиболее подходящая модель – Capability Maturity Model. Показана целесообразность ее усовершенствования путем добавления методик оценки трудоемкости проекта, идентификации основных типов ошибок, анализа причин этих ошибок. Для создания системы оценки использовались следующие инструменты: анализ Парето, каузальный анализ, подход с количеством вариантов использования (вариант использования описывает типичное взаимодействие между пользователем и системой), анализ и предупреждение возникновения ошибок конфигурирования программного обеспечения ИТ-компании. На базе усовершенствованной методики управления проектами ИТ-компании предложена концептуальная структура программного комплекса для оценки трудоемкости реализации проекта и поиска ошибок на стадии его выполнения. Система оценки создана с помощью среды разработки «1С Предприятие» и использует анализ таких видов объектов: архива проектов, реестра проектов, бюджетов и библиотеки требований.

Ключевые слова: ИТ-проект, жизненный цикл проекта, модель СММ, методика оценка трудоемкости, стадии проекта, идентификация основных типов ошибок, анализ причин ошибок, теоретико-множественная модель, концептуальная структура программного комплекса, принятие решений

Графическая аннотация (Graphical annotation)



IMPROVEMENT PROJECT MANAGEMENT TECHNIQUES IT COMPANY BASED MODEL CMM

The article has been received by editorial board 06.12.2016, in the final version – 26.01.2017.

Muntyanova Tatyana P., postgraduate student, Astrakhan State Technical University, 16 Tatishchev St., Astrakhan, 414056, Russian Federation, e-mail: orekhova.t.p@gmail.com

The set-theoretic model, reflecting activity of IT company, based on the life cycle of project, has been created. Also, is described the structural-functional model of project management. Different uniform approaches to IT project management has been studied, has been chosen the most suitable model – Capability Maturity Model. Has been showed the expediency of it improvement by means of the addition of methods of assessing the complexity of the project, identification of the main types of errors, the analysis of the causes of errors. For creation of the system have been used, Pareto analysis, causal analysis, the approach of using cases (use case describes a typical interaction between the user and the system), analysis and prevention of errors in configuration of software for IT companies. There is offered the conceptual framework of the software complex of bounds for the complexity of the project and search for errors on the stage of the project based on the advanced techniques of project management of IT company. The system is developed based on 1C Enterprise and use analysis of such types of objects as archive projects, registry of projects, budgets and library requirements.

Keywords: IT-project, project life cycle, Capability Maturity Model, methods of assessing the complexity of the project, identification of the main types of errors, the analysis of the causes of errors, set-theoretic model, conceptual framework of the software complex, making decisions

Введение. Термин «ИТ-проект» используется для обозначения деятельности, имеющей целью создание уникальных информационных продуктов, услуг или результатов [5], а также внедрения ранее выполненных ИТ-разработок. При этом около 90 % ИТ-проектов аналогичны уже ранее выполненным. Поэтому у руководителя проекта обычно имеется опыт реализации соответствующих задач по проектам и понимание возможных проблем [3]. Однако 10 % проектов – инновационные. Они реализуются «с нуля» и требуют нестандартных управленческих решений с высокими рисками. Это требует от руководителей ИТ-проектов глубоких знаний методики проектного управления и совершенствования её применения в сфере информационных технологий.

Известны различные унифицированные подходы к управлению ИТ-проектами, закрепленные в американском национальном стандарте Project Management Body of Knowledge (PMBoK) [18], российском стандарте «Национальные требования к компетентности специалистов по управлению проектами (НТК)» [10], в эволюционной модели Capability Maturity Model (CMM) [8] и др. Стандарты PMBoK и НТК регламентируют, «как» делать и представляют собой свод знаний по управлению проектами.

Модель CMM регламентирует, «что» делать. Поэтому она подходит для различных моделей жизненного цикла, методологий разработки и не накладывает каких-либо ограничений на применяемые в проектах стандарты документирования, инструментарий, среду разработки и языки, методы алгоритмизации [17]. Для многих ИТ-компаний этот фактор является решающим. При планировании и реализации ИТ-проектов, возникает целый ряд вопросов, связанных с определением и детальным структурированием необходимых работ, с оценкой трудоемкости работ по проекту, с управлением и контролем за исполняемыми работами. Поэтому целью данной статьи является совершенствование процессов управления ИТ-проектами, связанными с разработкой программного обеспечения на основе модели CMM – с учетом оценки трудоемкости проекта, идентификации основных типов ошибок, анализа причин этих ошибок.

Структурно-функциональная модель управления ИТ-проектом на основе CMM. ИТ-проекты согласно модели CMM проходят через ряд последовательных или перекрывающихся во времени этапов X , $X = \langle X0, X1, X2, X3, X4, X5, X6, X7, X8, X9 \rangle$ (рис. 1). При этом каждый этап фокусируется на подгруппе операций проекта и процессов управления проектом [4]. На этом рисунке в виде столбчатой диаграммы отражены этапы проекта (они показаны номерами) в отношении типичного соотношения времен и стоимостей – для внедрения компанией ИТ-проекта в условиях России.

Для формализованного описания модели управления проектами автором было выбрано теоретико-множественное представление, возможности которого близки к семантическим представлениям пользователя [9]. Соответственно деятельность ИТ-компаний по управлению проектами может быть представлена в виде теоретико-множественной модели, отражающей следующую совокупность множеств:

$$DIT = \langle CMM, X, RIT, P, M \rangle,$$

где CMM – стандарт, на основе которого разработана модель; OP – множество основных процессов, осуществляемых в рамках проектной деятельности ИТ-компаний; RIT – множество ресурсов ИТ-компаний, участвующих в управлении проектом; P – множество показателей эффективности управления проектами [13], M – методики, используемые для управления проектами.

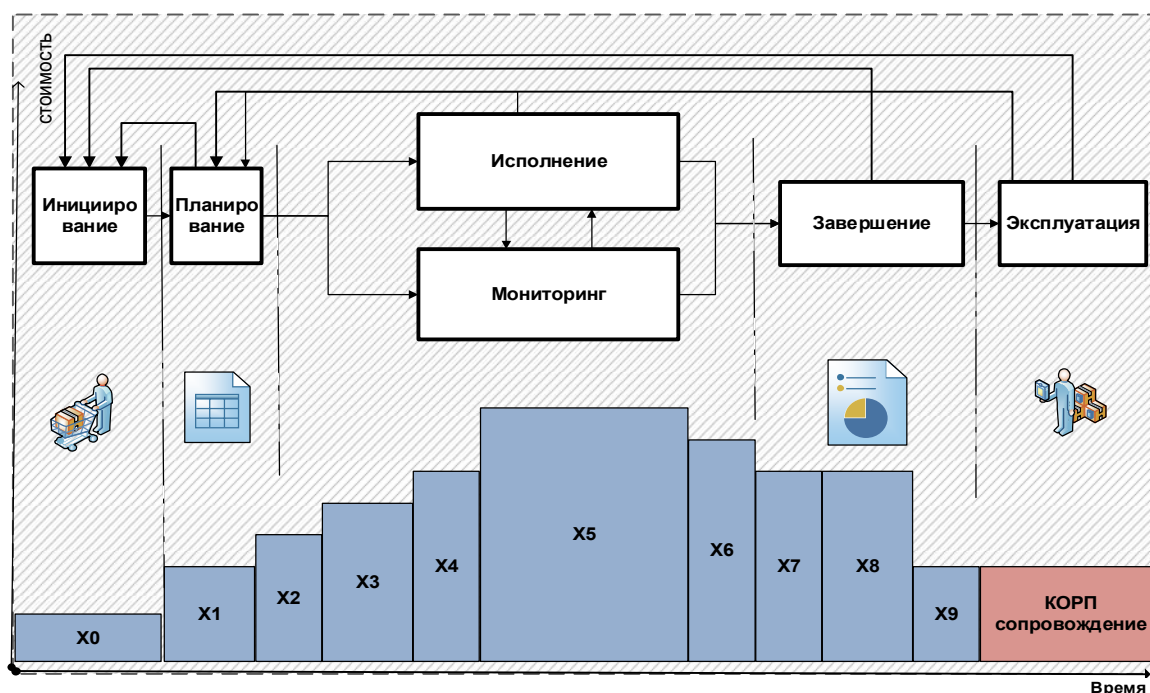


Рисунок 1 – Типичный системный жизненный цикл ИТ-проекта

Обозначения: X_0 – инициирование (инициация проекта); X_1 – обследование бизнес-процессов; X_2 – составление «Технического задания» (интеграция с системами); X_3 – составление «Технического проекта» (прототипирование); X_4 – составление «Программы и методики испытаний» (контрольные примеры); X_5 – разработка (разработка ПО, согласно функциональным требованиям, а также внутреннее и внешнее тестирование, нагрузочное тестирование, интеграционное тестирование); X_6 – конвертация данных в разработанную систему; X_7 – обучение персонала; X_8 – опытная эксплуатация; X_9 – перевод в промышленную эксплуатацию; корпоративное (КОРП) сопровождение-поддержка системы Заказчика

Каждое из множеств, входящих в модель, может быть представлено в виде набора элементов, связей, характеристик и совокупностей. Рассмотрим параметры каждого множества.

Множество этапов ИТ-проекта компании, можно представить в виде следующей совокупности:

$$X = \langle ep_i, se, zp, slpr \rangle,$$

где ep_i – этап проекта; se – срок этапа, zc – задача этапа проекта, $slpr$ – сложность этапа проекта.

Множество ресурсов ИТ-компании, участвующих в управлении проектом, можно представить в виде следующей совокупности:

$$RIT = \langle TR, ThR \rangle.$$

где TR – множество, определяющее трудовые ресурсы (здесь и далее фигурные скобки означают список объектов):

$$TR = \{kvr, or, krkv, zr\}$$

где kvr – квалификация сотрудника; or – опыт работы сотрудника; $krkv$ – количество трудовых ресурсов (сотрудника или группы сотрудников) данной квалификации, доступных для компании; zr – загрузка сотрудника (или группы сотрудников) по проектам.

ThR – множество, определяющее технические ресурсы ИТ-компании.

$$ThR = \{pro, er, sre\},$$

где pro – производительность; er – эргономичность, sre – срок эксплуатации.

Множество задач этапов проекта представим в виде следующей совокупности:

$$Zc = \{sl, sr, td, tz\},$$

где sl – сложность задачи; sr – срок выполнения задачи; td – трудоемкость задачи, tz – тип задачи.

Множество показателей эффективности управления проектом, представим в виде следующей совокупности:

$$P = \{r, k, s\},$$

где r – риски проекта, k – качество аппаратно-программного комплекса, используемого при разработке ИТ-проекта, s – плановые сроки.

Множество методик, совершенствующих CMM, представим в виде следующей совокупности:

$$M = \{Mo, Mp, Mk\},$$

где Mo – методика оценки трудоемкости, Mp – идентификация основных типов ошибок, Mk – каузальный анализ для выявления причин ошибок, определения решений, воздействующих на эти причины.

Проведена структуризация и формализация процесса управления ИТ-проекта на основе CMM (рис. 2). Определены структура и системные взаимосвязи между составляющими ИТ-проекта [11], включая методики, совершенствующие CMM за счет использования оценки трудоемкости, идентификации основных типов ошибок, анализа причин ошибок.

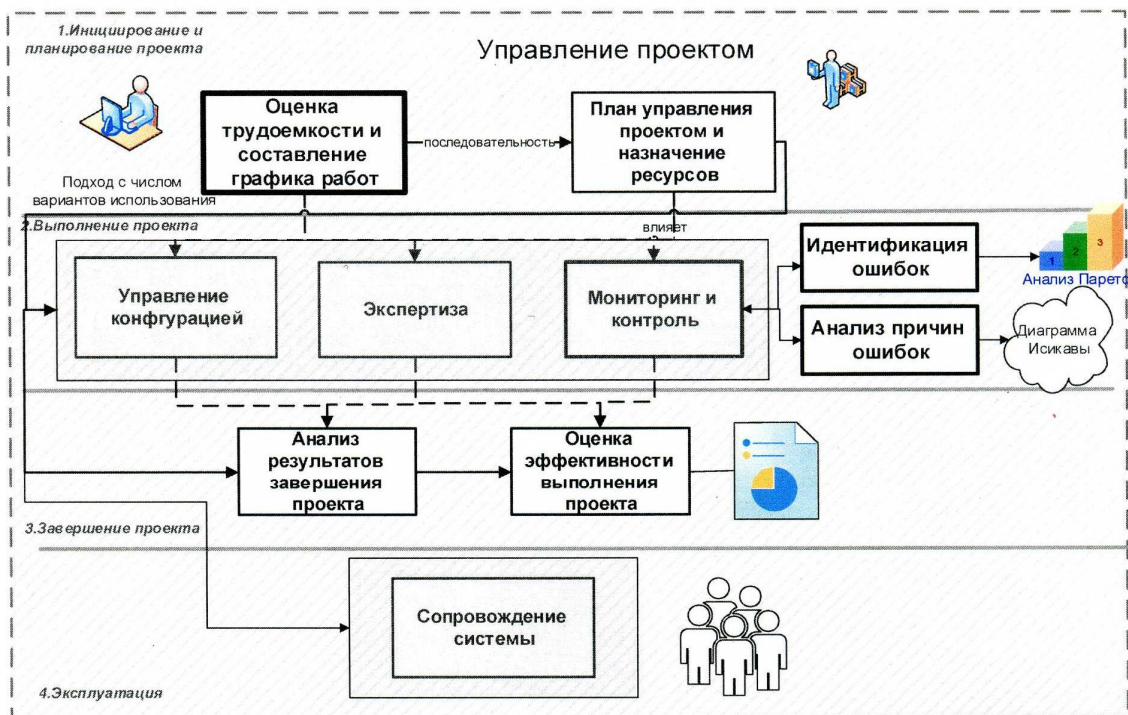


Рисунок 2 – Структурно-функциональная модель управления проектами

Оценка трудоемкости. Вопрос оценки трудоемкости ИТ-проекта по разработке программного обеспечения исследовался в работах Стенли И.Портни [2], Алана [1] и др.

Трудоемкость ИТ-проекта на основе линейной зависимости может быть оценена по формуле:

$$T = R \times P,$$

где R – размер исходного кода программного обеспечения (ПО); P – время работы, используемое алгоритмом (временная производительность).

Существует множество моделей, использующих для оценивания нисходящий подход. Из них наиболее известна модель СОСОМО [16]. Эта модель использует формулу регрессии с параметрами, определенными на основании данных, собранных по проекту.

При восходящем подходе сначала получают оценки для частей проекта, а потом – для проекта в целом. В этой стратегии изначально перечисляются основные действия, затем оценивается трудоемкость для каждого из действий. На основе этих оценок определяется трудоемкость для всего проекта.

Для обоих подходов (нисходящего и восходящего) нужна информация о проекте: размер (для нисходящего подхода) и перечень задач (для восходящего). Во многих отношениях эти подходы дополняют друг друга. Точность оценок в обоих случаях повышается, если появляется дополнительная информация о проекте или если проект уже выполняется и есть какие-то данные о ходе его выполнения. Например, оценить трудоемкость проекта крайне трудно, если заданы требования очень высокого уровня. Задача оценки упрощается, если окончено функционально-структурное проектирование, и еще более упрощается, если уже начата разработка кода. Недостатком обоих этих подходов является то, что точность оценок зависит от точки проекта по времени, в которой оценивается трудоемкость. При этом точность оценки возрастает по мере появления новой информации о проекте [14].

Для оценки трудоемкости реализации проекта был предложен подход с числом вариантов использования UCP (Use Case Points) [12]. Вариант использования описывает типичное взаимодействие между пользователем и системой. Подход UCP можно применить, если спецификация требований определяется с помощью вариантов использования.

Все варианты использования делятся на три типа: простые, средние и сложные в зависимости от количества транзакций в потоках событий (основных и альтернативных). В данном случае под транзакцией понимается атомарная последовательность действий, которая выполняется полностью или отменяется.

$$k = \begin{cases} 1, \text{ простой, если } \leq 3 \text{ транзакций} \\ 2, \text{ средний, если } 4 > \text{ транзакций} \geq 7, \\ 3, \text{ сложный, если } > 7 \text{ транзакций} \end{cases}$$

Значение UCP вычисляется следующим образом:

$$UCP = UUCP \times TCF \times EF, \quad (1)$$

где $UUCP$ – общий весовой показатель вариантов использования (Unadjusted Use Case Points). Он вычисляется по формуле:

$$UUCP = y_1 \times k_1 + y_2 \times k_2 + y_3 \times k_3, \quad (2)$$

где y_i – общее количество вариантов использования каждого типа (простого, среднего, сложного); k_i – коэффициент сложности.

TCF – техническая сложность проекта (Technical Complexity Factor) вычисляется по формуле:

$$TCF = 0.6 + (0.01 \times TFactor), \quad (3)$$

где

$$TFactor = \sum_{i=1}^I T_i V_i, \quad (4)$$

где I – порядковый номер показателя; T_i – показатель технической сложности; V_i – значение (вес) каждого показателя («0» означает отсутствие значимости показателя для данного проекта, «3» – наиболее высокую значимость).

EF – фактор окружения (Environmental Factor), вычисляется по формуле:

$$EF = 1.4 + (-0.03 \times FFactor), \quad (5)$$

где $FFactor$ вычисляется с учетом показателей, представленных в таблице 1.

$$FFactor = \sum_{i=1}^I F_i V_i, \quad (6)$$

где I – порядковый номер показателя; F_i – показатель «фактора окружения»; V_i – значение (вес) каждого показателя («0» означает отсутствие значимости показателя для данного проекта, «3» – наиболее высокую значимость).

Таблица 1 – Показатели «фактора окружения» и их значения

Факторы	Показатель
Уровень опыта	$F1 = \begin{cases} 0, \text{ отсутствие опыта} \\ 1, \text{ средний опыт,} \\ 2, \text{ эксперт.} \end{cases}$
Мотивация сотрудника	$F2 = \begin{cases} 0, \text{ отсутствие мотивации} \\ 1, \text{ средняя мотивация,} \\ 2, \text{ высокая мотивация.} \end{cases}$
Стабильность требований Заказчика к ИТ-проекту	$F3 = \begin{cases} 0, \text{ нестабильные требования,} \\ 1, \text{ средняя стабильность,} \\ 2, \text{ неизменные требования.} \end{cases}$
Сложность языка программирования, применяемого для ИТ-проекта	$F4 = \begin{cases} 1, \text{ легкий в использовании,} \\ 2, \text{ средний в использовании,} \\ 3, \text{ очень сложный язык.} \end{cases}$
Для сотрудников, занятых неполный рабочий день	$F5 = \begin{cases} 0, \text{ отсутствие сотрудников, занятых неполный день} \\ 1, \text{ половина сотрудников занята неполный день,} \\ 2, \text{ весь технический персонал занят неполный день.} \end{cases}$

Ошибки являются важным пунктом отслеживания на этапе разработки программного кода. Предупреждение ошибок заключается в сборе сведений об обнаруженных ошибках и в предупреждении возникновения ошибок в оставшейся части проекта. Действия по предупреждению ошибок обычно выполняются в проекте дважды: первый раз – когда программирование и тестирование элементов выполнено приблизительно для 20 % модулей, и еще раз – когда запрограммировано и протестировано 50 % модулей. Предупреждение ошибок включает выполнение анализа Парето для идентификации основных типов ошибок, каузального анализа для выявления причин ошибок, определения решений, воздействующих на эти причины.

Идентификация основных типов ошибок. Анализ Парето – это широко распространенный статистический прием, используемый для анализа причин и эффектов. Такой вид анализа является одним из основных инструментов управления качеством. Иногда его также называют правилом 80–20. Это правило гласит: 80 % проблем возникают из-за 20 % возможных источников. Для ПО это может означать, что 20 % коренных причин порождают 80 % ошибок, или что 80 % ошибок обнаруживаются в 20 % кода [6].

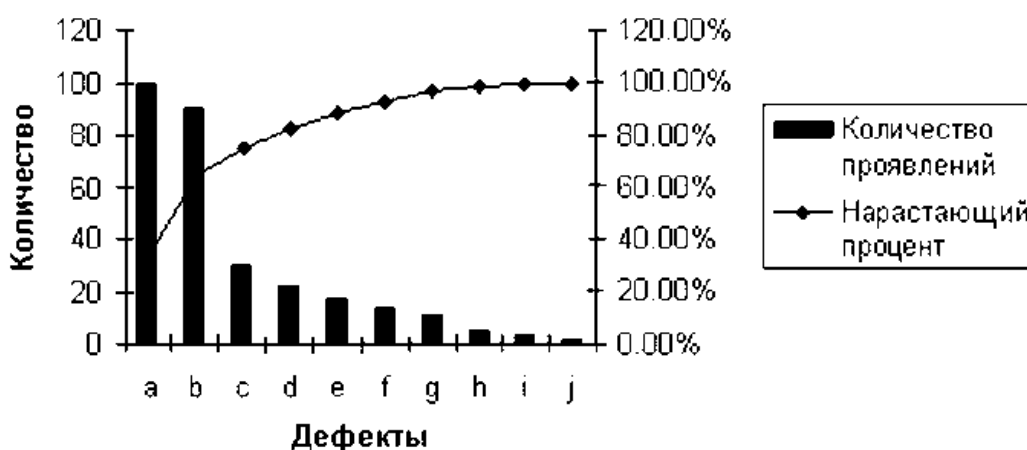


Рисунок 3 – Типичная диаграмма Парето для ошибок ИТ-проекта

Обозначения: *a* – логика; *b* – пользовательский интерфейс; *c* – стандарты; *d* – вопросы проектирования; *e* – граничные условия; *f* – согласованность; *g* – «жесткое» программирование (стиль программирования, при котором логика жестко завязана к данным); *h* – модульность; *i* – ошибки при тестировании; *j* – прочее

Наибольшее число ошибок в разрабатываемом ПО относится к типу синтаксических ошибок при записи операторов, но они устраняются в оперативном порядке и далее не рассматриваются. Следующими по частоте следуют логические ошибки, далее ошибки пользовательского интерфейса и ошибки стандартов. Суммарная доля для ошибок этих трех категорий составляет более 88 % от общего количества ошибок. Причем доля ошибок первых двух категорий превышает 75 % [7]. Поэтому очевидно, что предупреждение ошибок должно быть нацелено в первую очередь на эти две или три категории.

Диаграмма Парето помогает идентифицировать основные типы ошибок, которые были найдены и которые, если не предпринять специальных мер, с наибольшей вероятностью будут обнаружены в остальной части проекта. Эти ошибки можно считать «эффектами», которые требуется минимизировать в будущем. Чтобы сократить их число, необходимо найти основные причины и попытаться устранить их. При построении этой диаграммы в первую очередь выполняется идентификация анализируемого эффекта. По описанию выше такой эффект можно определить как «слишком много логических ошибок». Для того чтобы выявить его причины, нужно установить главные категории причин. В производстве такими главными причинами часто оказываются рабочая сила, машины, методы, материалы, измерения и условия производства.

Для каузального (причинно-следственного) анализа в ИТ-компании стандартный набор главных причин ошибок включает следующее: процесс, персонал, технологию и обучение (обучение отделено от персонала, поскольку эта причина проявляется очень часто). В основной структуре диаграммы эффект показывается прямоугольником справа. От прямоугольника проводится прямая горизонтальная линия, а наклонные линии соединяют каждую главную причину с основной горизонтальной линией.

Для сокращения числа основных типов ошибок необходимо найти и устранить их основные причины. Для определения причин наблюдаемых эффектов можно использовать диаграмму причин и эффектов (диаграмму причинно-следственных связей). Анализ причин ошибок должен осуществляться командой предупреждения ошибок проекта.

Анализ причин ошибок. При анализе главных причин для каждой из них нужно поставить ключевой вопрос: «Почему эта причина производит такой эффект?» Ответы на эти вопросы (для основных причин) становятся подпричинами. Их изображают короткими горизонтальными линиями, соединенными

с линией, представляющей главную причину. Затем тот же вопрос задается по поводу идентифицированных (выявленных) подпричин. Этот процесс «почему – почему – почему» повторяется до тех пор, пока не идентифицированы все коренные причины, т.е. те причины, для которых вопрос уже не имеет смысла. Когда на диаграмме отмечены все причины, итоговая картинка напоминает скелет рыбы. Поэтому диаграмму причин и эффектов также называют диаграммой «рыбьего скелета» или диаграммой Исикавы (*Ishikawa*), по имени ее создателя (разработчика) [7]. Диаграмма «рыбьего скелета» позволяет идентифицировать все причины исследуемого эффекта. Обычно первоначальная диаграмма содержит слишком много причин. Однако, очевидно, что некоторые из них оказывают большее влияние, чем другие. Следовательно, в конце анализа коренных причин, главным образом в процессе обсуждения, необходимо идентифицировать несколько основных причин. Менее значимые причины из диаграммы Исикавы обычно исключаются для облегчения ее восприятия (повышения наглядности).

Для предупреждения ошибок иногда можно, провести все исследование для одной – двух основных категорий ошибок, определенных в анализе Парето.

Диаграмма «рыбий скелет» для анализа ошибок в ИТ-проекте представлена на рисунке 4.



Рисунок 4 – Диаграмма «рыбий скелет» для ИТ-проекта

В данном анализе командой предупреждения ошибок с помощью метода мозгового штурма рассматриваются три главных типа ошибок:

- ошибки логики;
- ошибки пользовательского интерфейса;
- ошибки стандартов.

Например, эффект был определен как «слишком много ошибок логики». Проанализировав: «Почему персонал вносит в разработку так много ошибок логики?», командой были идентифицированы некоторые (почти очевидные) причины: отсутствие (или недостаточность) обучения, недосмотр (т.е. недостаточное внимание при работе), отсутствие технических навыков. Аналогично, когда команда пыталась понять: «Почему процессы разработки ПО вызывают столь много ошибок логики?», то получила ответы: «Стандарты неполно документированы» и «Персонал не знает стандартов». Для «технологии» причинами были: «непонятные спецификации» и «технические проблемы с инструментами». Метод мозгового штурма для каузального анализа позволяет выявить гораздо больше причин. После получения перечня всех предположений, сделанных в ходе совещания, команда предупреждения ошибок назначает им приоритеты, рассматривает каждую ошибку и идентифицирует ее причины. Наиболее часто проявляются причины, имеющие наивысшие приоритеты, которые изображены на рисунке 4.

В целом процедура выполнения методики анализа и предупреждения возникновения ошибок проектов в ИТ-компаниях состоит из следующих этапов:

- 1) составление списка всех идентифицированных (выявленных) ошибок к моменту проведения мозгового штурма;
- 2) сортировка ошибок по типам в порядке снижения их количества;

- 3) вычисление в процентах доли каждого типа ошибок от общего числа обнаруженных ошибок;
- 4) идентификация типа ошибки, составившей около 80 % от общего числа ошибок с помощью анализа Парето (если ошибка такого типа имеет место);
- 5) построение диаграммы «рыбьего скелета»;
- 6) определение главных категорий причин. Они могут быть как стандартными, так и несколько измененными – с учетом особенностей проблемы;
- 7) запись главных категорий в прямоугольники и соединение прямоугольников с «хребтом» (основной горизонтальной линией) наклонными стрелками. Тем самым будут созданы основные «кости» диаграммы «рыбий скелет»;
- 8) определение подпричин для главных причин. При этом для каждой главной причины задается один и тот же вопрос: «Почему эта главная причина производит данный эффект?»;
- 9) добавление в диаграмму подпричин. Они группируются вокруг «кости», соответствующей главной причине. При необходимости продолжить разбиение причин или подпричин на более низкие иерархические уровни.

Следует также отметить, что использование программных средств поддержки разработки программ, а также соблюдение «дисциплины программирования» обычно приводит к снижению количества ошибок.

Усовершенствованная методика управления проектами ИТ-компаний на основе модели СММ. Такая методика предполагает использование следующих решений: модель зрелости создания программного обеспечения СММ, подход оценки трудоемкости, анализ идентификации и устранения причин ошибок.

При этом используется следующий алгоритм:

- 1) составление плана-графика работ ИТ-проекта;
- 2) определение весовых показателей вариантов использования UUCP по формуле 2;
- 3) определение технической сложности проекта (TCF) по формуле 3;
- 4) определение фактора окружения (EF) по формуле 5;
- 5) вычисление окончательного значения UCP по формуле 1;
- 6) сбор сведений об обнаруженных ошибках для предупреждения возникновения ошибок в оставшейся части проекта;
- 7) использование диаграммы Парето для определения причин наблюдаемых эффектов (ошибок);
- 8) исследование трех категорий, определенных в анализе Парето (ошибки логики, ошибки пользовательского интерфейса, ошибки стандартов);
- 9) идентификация причин ошибок с помощью диаграммы «рыбий скелет» – каузальный анализ (анализ причинно-следственных связей).

С обеспечения поддержки принятия и реализации решений в рамках использования приведенного выше алгоритма целесообразно применение программных средств. На базе усовершенствованной методики управления проектами ИТ-компаний предложена концептуальная структура программного комплекса для оценки трудоемкости проекта и поиска ошибок на стадии его выполнения. На рисунке 5 представлена концептуальная структура такого программного комплекса.

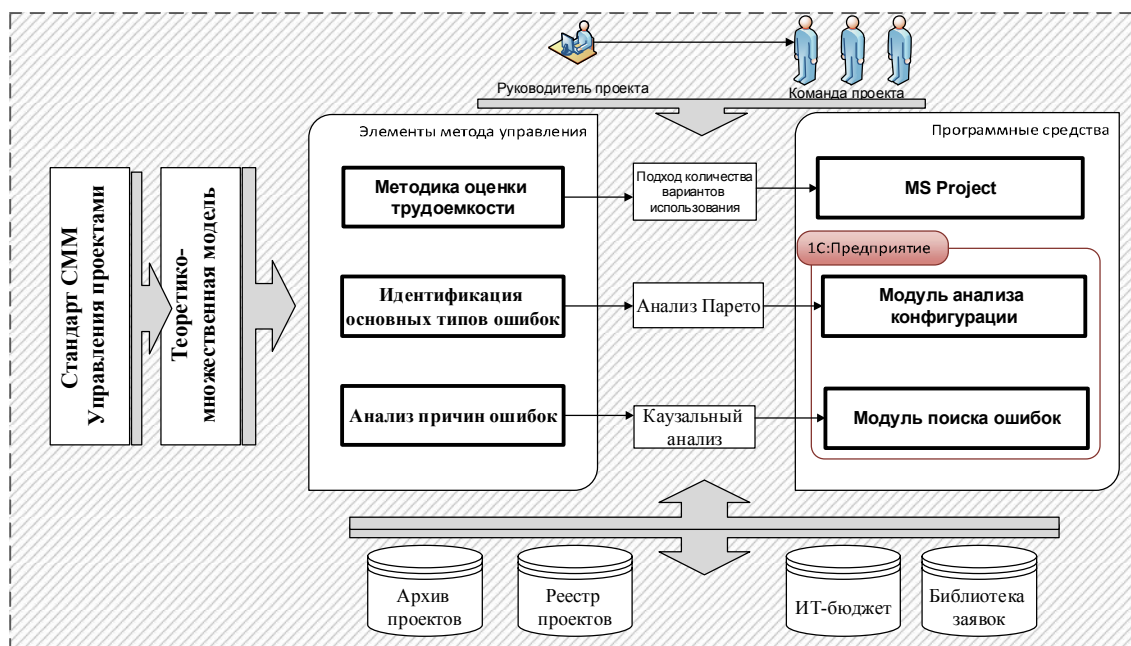


Рисунок 5 – Концептуальная структура программного комплекса управления ИТ-проектами

В комплексе реализованы следующие методы: анализ Парето, каузальный анализ [7], подход на основе количества вариантов использования [12]. Комплекс создан с помощью среды разработки «1С Предприятие». В нем используется анализ таких объектов: архива проектов, реестра проектов, бюджетов проектов; библиотеки требований (это совокупности требований к проекту, исполнение которых минимизирует риски ИТ-проекта). Для отражения всех аспектов проекта, используется «архив проекта», представляющий собой комплект документов. Информация, необходимая для эффективного управления проектом хранится в «реестре проектов».

Эффективность предложенных методов и созданного автором программного обеспечения подтверждается практикой использования разработки в ИТ-компании ООО ПКФ «Бест Софт». При этом повышение эффективности выполнения ИТ-проектов составило около 45 % [15].

Выводы. Проведен системный анализ деятельности ИТ-компании, подходов к управлению ИТ-проектами. Поставлена задача совершенствования методики управления проектами ИТ-компании на основе использования модели СММ. Предложенная нами усовершенствованная методика управления проектами ИТ-компании, включает в себя следующее: теоретико-множественную модель управления проектами, отражающую деятельность ИТ-компании; оценку трудоемкости проекта с использованием подхода с числом вариантов использования; анализ и предупреждение возникновения ошибок конфигурирования программного обеспечения ИТ-компании. Итоговым результатом работы является научно обоснованное решение задачи повышения качества управления проектами ИТ-компании, обладающее определенными функциональными преимуществами по сравнению с существующими разработками.

Список литературы

1. Алан Д. Опп. Управление проектами. Руководство по ключевым процессам, моделям и методам / Алан Д. Опп. – Баланс Бизнес Букс, 2006. – 224 с.
2. Архипенков С. Руководство командой разработчиков программного обеспечения. Прикладные мысли / С. Архипенков. – Москва, 2008. – 80 с.
3. Белов А. Г. Система ключевых показателей эффективности для проекта разработки тиражного программного обеспечения / А. Г. Белов, А. Г. Кравец // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 4. – С. 11–22.
4. Брумштейн Ю. М. Сравнительный анализ функциональности программных средств управления проектами, распространяемых по модели SAAS / Ю. М. Брумштейн, И. А. Дюдинов // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 4. – С. 34–51.
5. Грекул В. И. Методические основы управления ИТ-проектами : учебник / В. И. Грекул, Н. Л. Коровкина, Ю. В. Куприянов. – Москва : Интернет Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2010. – 391 с.
6. Заргарян Ю. А., Васильев И. А. Аспекты нахождения Парето-оптимального решения при нечетком описании исходных данных / Ю. А. Заргарян, И. А. Васильев // Инновации в сельском хозяйстве. – 2012. – № 2 (2). – С. 54–59.
7. Колосова Е. А. Каузальный анализ / Е. А. Колосова // Тезаурус социологии / Ж.Т. Тощенко, Е. А. Гришина, В. Ф. Левичева, А. В. Кученкова ; под ред. Ж. Т. Тощенко. – Москва, 2013. – С. 196–197.
8. Паулк Марк. Модель зрелости процессов разработки программного обеспечения / Паулк Марк, Билл Куртис, Мэри Бет Хриссис, Чарльз В. Вебер, Сьюзен М. Гарсия, Мерилин Буш ; пер. В. Рябикина. – CMMI Product Team, 2010. – 123 с.
9. Проталинский О. М. Теоретико-множественная модель процессов грузового порта / О. М. Проталинский, А. А. Ханова, И. О. Григорьева // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2009. – № 2. – С. 83–89.
10. Управление проектами: Основы профессиональных знаний, Национальные Требования к Компетентности специалистов (NCB – SOVNET National Competence Baseline Version 3.0). – Москва : Проектная практика, 2010. – 256 с.
11. Ханова А. А. Системные взаимосвязи стратегического управления и моделирования социально-экономических систем на основе сбалансированной системы показателей / А. А. Ханова, А. С. Хортонен, Л. В. Парамзина // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2014. – № 2. – С. 109–116.
12. Ханова А. А. Оценка эффективности деятельности организации на основе сбалансированной системы показателей и имитационного моделирования (на примере грузового порта) / А. А. Ханова, И. О. Григорьева, Е. С. Потапова // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – 2009. – Т. 6, № 91. – С. 119–126.
13. Ханова А. А. Организация принятия решений в виде цикла управления эффективностью организации / А. А. Ханова, А. С. Пономарёва // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2011. – № 2. – С. 171–177.
14. Шанченко Н. И. Оценка трудоемкости разработки программного продукта : методические указания / Н. И. Шанченко. – Ульяновск : Ульяновский государственный технический университет, 2015. – 40 с.
15. Щербатов И. А. Математические модели сложных слабоформализуемых систем: компонентный подход / И. А. Щербатов // Системы. Методы. Технологии. – 2014. – № 2 (22). – С. 70–78.
16. Boehm Barry. Software engineering economics / Boehm Barry. – Englewood Cliffs, NJ : Prentice-Hall, 1981.
17. Jalote P. CMM in Practice: Processes for Executing Software Projects at Infosys / P. Jalote. – Addison-Wesley, 2010.
18. Project Management Institute, Inc. PMBOK. Руководство к Своду знаний по управлению проектами. – 5-е изд. – 2013. – 241 с.

References

1. Alan D. Orr. *Upravlenie proektami. Rukovodstvo po klyuchevym protsessam, modelyam i metodam* [Project management. Guidelines for key processes, models and methods], Balans Biznes Buks Publ., 2006. 224 p.
2. Arkhipenkov S. *Rukovodstvo komandoy razrabotchikov programmnogo obespecheniya. Prikladnye mysli* [Management software development team. applied thought], Moscow, 2008. 80 p.
3. Belov A. G., Kravets A. G. Sistema klyuchevykh pokazateley effektivnosti dlya proekta razrabotki tirazhnogo programmnogo obespecheniya [System key performance indicators for project development retail software]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2014, no. 4, pp. 11–22.
4. Brumshteyn Yu. M., Dyudikov I. A. Sravnitelnyy analiz funktsionalnosti programmykh sredstv upravleniya proektami, rasprostranyaemykh po modeli SAAS [Comparative analysis of functionality software project management released under the SAAS model]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2014, no. 4, pp. 34–51.
5. Grekul V. I., Korovkina N. L., Kupriyanov Yu. V. *Metodicheskie osnovy upravleniya IT-proektami* [Methodical bases of management of IT projects], Moscow, Internet Universitet Informatsionnykh Tehnologiy: BINOM. Laboratoriya znaniy Publ., 2010. 391 p.
6. Zargaryan Yu. A., Vasilev I. A. Aspekty nakhozheniya Pareto-optimalnogo resheniya pri nechetkom opisani iskhodnykh dannykh [Aspects of finding Pareto-optimal solutions with a fuzzy description of the source data]. *Innovatsii v selskom hozyaystve* [Innovations in agriculture], 2012, no. 2 (2), pp. 54–59.
7. Kolosova Ye. A. Kauzalnyy analiz [Causal analysis]. *Tezaurus sotsiologii* [Thesaurus of Sociology], Moscow, 2013, pp. 196–197.
8. Paulk Mark, Bill Kurtis, Mjeri Bet Hrisiss, Charl'z V. Veber, S'juzen M. Garsija, Merilin Bush. *Model zrelosti protsessov razrabotki programmnogo obespecheniya* [Model maturity of software development processes], CMMI Product Team Publ., 2010. 123 p.
9. Protalinskiy O. M., Khanova A. A., Grigoreva I. O. Teoretiko-mnozhestvennaya model protsessov gruzovogo porta [Set-theoretic model of the cargo port processes]. *Vestnik Astrahanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of the Astrakhan State Technical University. Series: Management, Computer Science and Informatics], 2009, no. 2, pp. 83–89.
10. *Upravlenie proektami: Osnovy professionalnykh znaniy, Natsionalnye Trebovaniya k Kompetentnosti spetsialistov (NCB – SOVNET National Competence Baseline Version 3.0)* [Project Management: The Basics of professional knowledge, to the national competent professionals Requirements (NCB – SOVNET National Competence Baseline Version 3.0)], Moscow, Proektnaya praktika Publ., 2010. 256 p.
11. Khanova A. A., Khortonen A. S., Paramzina L. V. Sistemnye vzaimosvyazi strategicheskogo upravleniya i modelirovaniya sotsialno-ekonomicheskikh sistem na osnove sbalansirovannoy sistemy pokazateley [System of strategic management of the relationship and simulation of socio-economic systems based on the Balanced Scorecard]. *Vestnik Astrahanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of the Astrakhan State Technical University. Series: Management, Computer Science and Informatics], 2014, no. 2, pp. 109–116.
12. Khanova A. A., Grigoreva I. O., Potapova Ye. S. Otsenka effektivnosti deyatel'nosti organizatsii na osnove sbalansirovannoy sistemy pokazateley i imitatsionnogo modelirovaniya (na primere gruzovogo porta) [Evaluating the effectiveness of the organization based on a balanced scorecard and simulation (in cargo port example)]. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikatsii. Upravlenie* [Scientific and Technical Journal of the St. Petersburg State Polytechnic University. Computer Science. Telecommunication and Control Systems], 2009, vol. 6, no. 91, pp. 119–126.
13. Khanova A. A., Ponomareva A. S. Organizatsiya prinyatiya resheniy v vide tsikla upravleniya effektivnostyu organizatsii [Organization of the decision-making in the form of organization performance management cycle]. *Vestnik Astrahanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of the Astrakhan State Technical University. Series: Management, Computer Science and Informatics], 2011, no. 2, pp. 171–177.
14. Shanchenko N. I. *Otsenka trudoemkosti razrabotki programmnogo produkta* [Evaluation of the complexity of software development products], Ulyanovsk, Ulyanovsk State Technical University publ. House, 2015. 40 p.
15. Shcherbatov I. A. Matematicheskie modeli slozhnykh slaboformalizuemyykh sistem: komponentnyy podkhod [Mathematical models of complex systems poorly formalized: Component Approach]. *Sistemy. Metody. Tekhnologii* [Systems. Methods. Technologies], 2014, no. 2 (22), pp. 70–78.
16. Boehm Barry. *Software engineering economics*, Englewood Cliffs, NJ. Prentice-Hall Publ., 1981.
17. Jalote P. *CMM in Practice: Processes for Executing Software Projects at Infosys*, Addison-Wesley, 2010.
18. *Project Management Institute, Inc. PMBOK. Rukovodstvo k Svodu znaniy po upravleniyu proektami* [Project Management Body of Knowledge], 5th ed. 2013. 241 p.